CLAIMS

What is claimed is:

1. An output engine operable to convert a file from a first format to a second format, the output engine comprising:

a decomposer operable to be called by a calling application and to receive a file and a desired file format from the calling application, the decomposer operable to decompose the file into a component architecture;

a writer operable to call the decomposer multiple times to retrieve the component architecture of the file and generate a new version of the file in the second format.

2. An output engine as claimed in claim 1, further comprising a second writer operable to call the decomposer multiple times to retrieve the component architecture of the file and generate a new version of the file in a third format.

3. An output engine as claimed in claim 1, wherein the decomposer includes a job processor.

4. An output engine as claimed in claim 3, wherein the job processor is operable to load pages of the file and associate data with each page.

5. An output engine as claimed in claim 3, wherein the decomposer includes a writer interface.

6. An output engine as claimed in claim 5, wherein the decomposer includes a calling application interface.

7. An output engine as claimed in claim 3, wherein the writer includes a job processor interface.

8. An output engine as claimed in claim 7, wherein the writer includes a layer processor, a collection processor, and an item processor.

9. An output engine as claimed in claim 8, wherein the writer includes a stack and an output module.

10. A method of converting an input file from a first format to a second format, the method comprising:

delivering a desired file format and the input file to a decomposer;

decomposing the input file into a component architecture in the decomposer; and

generating a new version of the input file in the second format by calling the decomposer multiple times from a writer.

11. A method of converting an input file as claimed in claim 10, the method further comprising:

sending the new version of the input file to a second writer; and

generating a new version of the input file in a third format.

12. A method of converting an input file as claimed in claim 10, the method further comprising converting the input file from a fourth format to a common file format prior to delivering the input file to a decomposer.

13. A method of converting an input file having at plurality of pages and formatted in a first format to an output file formatted in a second format, the method comprising:

receiving a file conversion request from a calling application;

loading each page of the input file, one page at a time, in a decomposer;

associating data with one or more of the plurality of pages;

decomposing objects in each page into a component architecture;

driving each page to a writer; and

generating the output file by calling a decomposer multiple times from a writer.

14. A method as claimed in claim 13, further comprising:

executing a plurality of writers in a chained fashion.

15. A method as claimed in claim 13, further comprising

sending the output file to a second writer; and

generating a new version of the output file in a third format.

16. A method as claimed in claim 13, further comprising

determining layers, collections, and items in each page;

pushing the determined layers, collections, and items onto a stack; and

assembling the determined layers, collections, and items into the output file.

17. A file conversion system comprising:

a workstation having a source application, an output engine, and a document manager, the output engine including

a decomposer operable to be called by a calling application and to receive a file and a desired file format from the calling application, the decomposer operable to decompose the file into a component architecture; and

a writer operable to call the decomposer multiple times to retrieve the component architecture of the file and generate a new version of the file in the second format.

a form data database and a form database, each accessible to the workstation; and

a server accessible to the workstation and having a document control and production engine.

18. A system as claimed in claim 17, wherein the output engine further comprises a second writer operable to call the decomposer multiple times to retrieve the component architecture of the file and generate a new version of the file in a third format.

19. A system as claimed in claim 17, wherein the decomposer includes a job processor.

20. A system as claimed in claim 19, wherein the job processor is operable to load pages of the file and associate data with each page.

21. A system as claimed in claim 19, wherein the decomposer includes a writer interface.

22. A system as claimed in claim 21, wherein the decomposer includes a calling application interface.

23. A system as claimed in claim 19, wherein the writer includes a job processor interface.

24. A system as claimed in claim 17, wherein the writer includes a layer processor, a collection processor, and an item processor.

25. A system as claimed in claim 17, wherein the writer includes a stack and an output module.

26. A system as claimed in claim 17, further comprising a converter accessible to the workstation and operable to convert files from a foreign format to a common format.